Amendment to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims

1-31. (Canceled)

32. (Currently amended): A method of transforming data, the method comprising:

positioning a definition pointer to point at a first compound transform definition within a transform definition file;

invoking a first parallel processing thread to read the pointed at first compound transform definition;

searching data to be transformed for a data element to be transformed, the search being responsive to the first compound transform definition;

calling a dynamic function defined in the transform definition file, the dynamic function located elsewhere in the transform definition file from the definition pointer position;

transforming any found data element into an output data file, responsive to the first compound transform definition and called dynamic function, a data structure of the output data file being responsive to a data structure of the first compound transform definition;

determining a type of the read first compound transform definition;

based on a determination that the first compound transform definition is compound, processing each sub-definition of the read first compound transform definition by repeating the positioning, invoking, searching, calling, and transforming for each sub-definition in order to transform each sub-definition recursively, wherein the data elements transformation includes nesting of a data element;

determining if all sub-definitions of the first compound transform definition have been processed;

positioning a definition pointer to point at a second compound transform definition within the transform definition file;

invoking a second parallel processing thread to read the pointed at second compound transform definition;

searching data to be transformed for another data element to be transformed, the search being responsive to the second compound transform definition;

transforming any found data element into the output data file, responsive to the second compound transform definition, the data structure of the output data file being responsive to the data structure of the second compound transform definition; and

if no data element is found to be transformed, adding one or more output data elements to the output data file responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element.

33-36. (Canceled)

37. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a value parameter configured to specify a value for inclusion in the output data file.

38. (Original): The method of claim 32, wherein the data element is a compound data element and the read transform definition includes a source record parameter configured to specify the compound data element.

39. (Previously presented): The method of claim 32, wherein the read first compound transform definition is in a meta-language format.

40. (Original): The method of claim 32, wherein the data to be transformed data is in a meta-language data format.

41. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a transform element having an output field name and a source field parameter.

42. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a value parameter configured to populate a field in the output data file.

43. (Canceled)

44. (Currently amended): A method of transforming data, the method comprising:

positioning a definition pointer to point at a first compound transform definition, the first compound transform definition being within a transform definition file;

invoking a first parallel processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition;

positioning a first payload pointer to point at a first data element to be transformed, the positioning of the first payload pointer being responsive to a data structure of the first compound transform definition;

calling a dynamic function defined in the transform definition file, the dynamic function located elsewhere in the transform definition file from the definition pointer position;

transforming the first data element into an output data file, responsive to the read first compound transform definition and called dynamic function;

determining a type of the read first compound transform definition;

based on a determination that the first compound transform definition is compound, processing each sub-definition of the read first compound transform definition by repeating the positioning, invoking, positioning, calling, and transforming for each sub-definition in order to transform each sub-definition recursively, wherein the data elements transformation includes nesting of a data element;

determining if all sub-definitions of the first compound transform definition have been processed;

positioning the definition pointer to point at a second compound transform definition, the second compound transform definition being within the transform definition file;

invoking a second parallel processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition;

positioning a second payload pointer to point at a second data element to be transformed, the positioning of the second payload pointer being responsive to a data structure of the second compound transform definition;

transforming the second data element into the output data file, responsive to the read second compound transform definition; and

if no first or second data element is found to be transformed, adding one or more output data elements to the output data file responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element.

45-47. (Canceled)

48. (Original): The method of claim 44, further including determining if all sub-elements of a compound element have been transformed and, if the determination returns a value of YES, returning to a calling process.

49. (Canceled)

50. (Previously presented): The method of claim 44, further including un-nesting the data element to be transformed.

51. (Previously presented): The method of claim 44, wherein the read first compound transform definition includes a source field parameter configured to specify the data element.

52. (Previously presented): The method of claim 44, wherein the read first compound transform definition includes a source record parameter configured to specify the compound data element.

53-55. (Canceled)

56. (Previously presented): The method of claim 44, wherein the transform definition file includes a tree data structure.

57-58. (Canceled)

59. (Currently Amended): A computer readable storage media having embodied thereon data, the data comprising:

computer instructions configured to position a definition pointer to point at a first compound transform definition, the first compound transform definition being within a transform definition file;

computer instructions configured to invoke a first parallel processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition;

computer instructions configured to position a first payload pointer to point at a first data element to be transformed, the positioning being responsive to a data structure of the first compound transform definition;

computer instructions configured to call a dynamic function defined in the transform definition file, the dynamic function located elsewhere in the transform definition file from the definition pointer position;

computer instructions configured to transform the first data element into an output data file, responsive to the read first compound transform definition and called dynamic function;

computer instructions configured to determine a type of the read first compound transform definition;

based on a determination that the first compound transform definition is compound, computer instructions configured to process each ~~sub- definition~~ sub-definition of the read first compound transform definition by repeating the positioning, invoking, positioning, calling, and transforming for each sub-definition in order to transform [[the]] each sub-definition recursively, wherein the data elements transformation includes nesting;

computer instructions configured to determine if all sub-definitions of the first compound transform definition have been processed;

computer instructions configured to position a second payload pointer to point at a second data element to be transformed, the positioning being responsive to a data structure of the second compound transform definition;

computer instructions configured to invoke a second parallel processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition; [[and]]

computer instructions configured to transform the second data element into the output data file, responsive to the read second compound transform definition; and

if no first or second data element is found to be transformed, computer instructions configured to add one or more output data elements to the output data file responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element.

60-64. (Canceled)

65. (Currently amended): An application system comprising:

a computing device comprising:

a memory; and

at least one processor operatively coupled to the memory, the at least one processor configured to:

position a definition pointer to point at a first compound transform definition within a transform definition file;

invoke a first parallel processing thread to read the pointed at first compound transform definition;

search data to be transformed for a data element to be transformed, the search being responsive to the first compound transform definition;

call a dynamic function defined in the transform definition file, the dynamic function located elsewhere in the transform definition file from the definition pointer position;

transform any found data element into an output data file, responsive to the first compound transform definition and called dynamic function, a data structure of the output data file being responsive to a data structure of the first compound transform definition;

determine a type of the read first compound transform definition;

based a on determination that the first compound transform definition is compound, process each sub-definition of the read first compound transform definition by repeating the positioning, invoking, searching, calling, and transforming for each sub-definition in order to transform the each sub-definition recursively, wherein the data elements transformation includes nesting of a data element;

determine if all sub-definitions of the first compound transform definition have been processed;

position a definition pointer to point at a second compound transform definition within the transform definition file;

invoke a second parallel processing thread to read the pointed at second compound transform definition;

search data to be transformed for another data element to be transformed, the search being responsive to the second compound transform definition;

transform any found data element into the output data file, responsive to the second compound transform definition, the data structure of the output data file being responsive to the data structure of the second compound transform definition; and

if no data element is found to be transformed, add one or more output data elements to the output data file responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element.

66. (Previously presented): The application system of claim 65, wherein the at least one processor is further configured to select the transform definition file from a set of transform definitions files, responsive to data associated with the data to be transformed.

67. (Canceled)

68. (Previously presented): The application system of claim 65, wherein the at least one processor is further configured to add data to the output data file, the added data being configured responsive to the transform definition file and having no contribution from the data to be transformed.